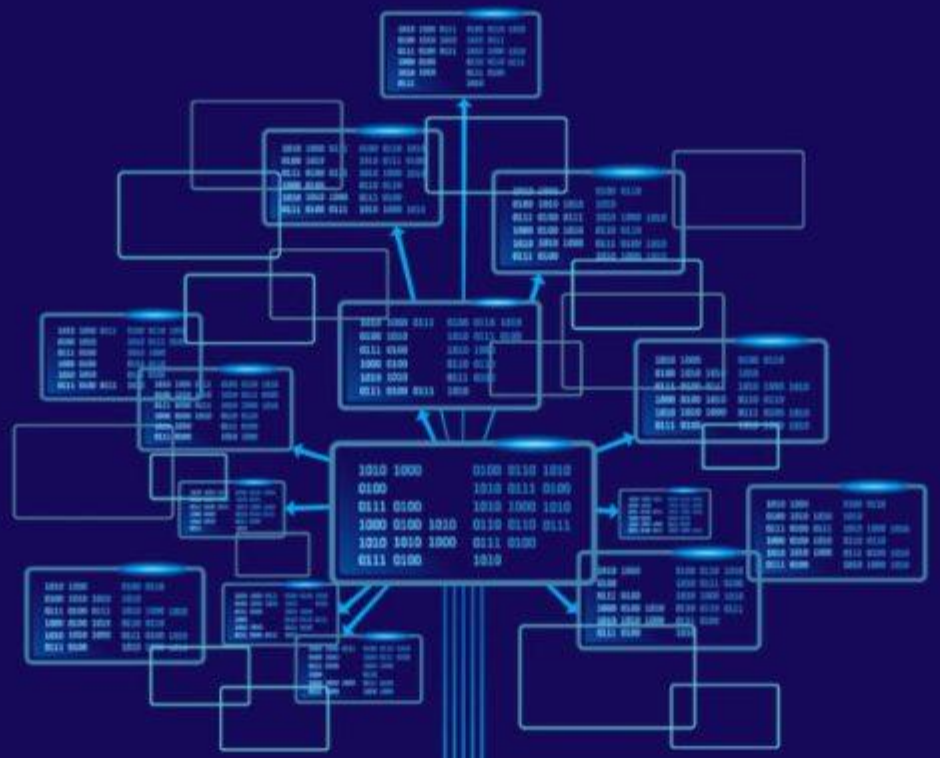


AĞAÇ ALGORİTMALARI

Dr. Öğr. Üyesi Nilgün Şengöz

Veri Madenciliğinde Karar Ağaçları

- Karar ağacı modeli adından da anlaşılacağı üzere ağaç görünümünde olan ve kullanıcıya sınıflama, kümeleme ve tahminler yapmada yardımcı bir modeldir
- Karar ağaçlarının oluşturulması basit, yorumlanmasının kolay olması ve veri tabanlarına kolaylıkla bütünleşerek tahminleme yapmasından dolayı oldukça sık kullanılan bir yöntemdir. Sınıflandırma modelleri arasında en çok tercih edilen model karar ağacıdır

Veri Madenciliğinde Karar Ağaçları

- Veri madenciliği kavramı çok sayıda ve büyük yapıdaki veri ambarları ve veri tabanlarının içerisindeki verilerin anlamlandırılması ve ilişkiler kurulmasına yardımcı olan istatistiki algoritmaları ve yapay zeka teknolojilerini kullanan bir yöntemdir.
- Veri arama tekniği olarak da isimlendirilmektedir. Veri madenciliği sürecinde beş adımdan oluşmaktadır
  1. Problemin tanımlanması
  2. Verilerin hazırlanması
  3. Modelin kurulması ve değerlendirilmesi
  4. Modelin kullanılması
  5. Modelin izlenmesi

Veri Madenciliğinde Karar Ağaçları

- Veri Madenciliği teknikleri, genellikle büyük ve çok sayıda verinin bulunduğu, bu verilerin eğitilmesi ve bu verilerden tahminleme işlemleri yapılmasında kullanılmaktadır.
- Karar ağaçları önemli sınıflama araçlarından birini oluşturmaktadır.
- Yapının öğrenmesi kolaydır ve bilgiler anlaşılır şekilde gösterilebilme özelliğine sahip olması karar vericiler için birtakım avantaj sunmaktadır.

Veri Madenciliğinde Karar Ağaçları

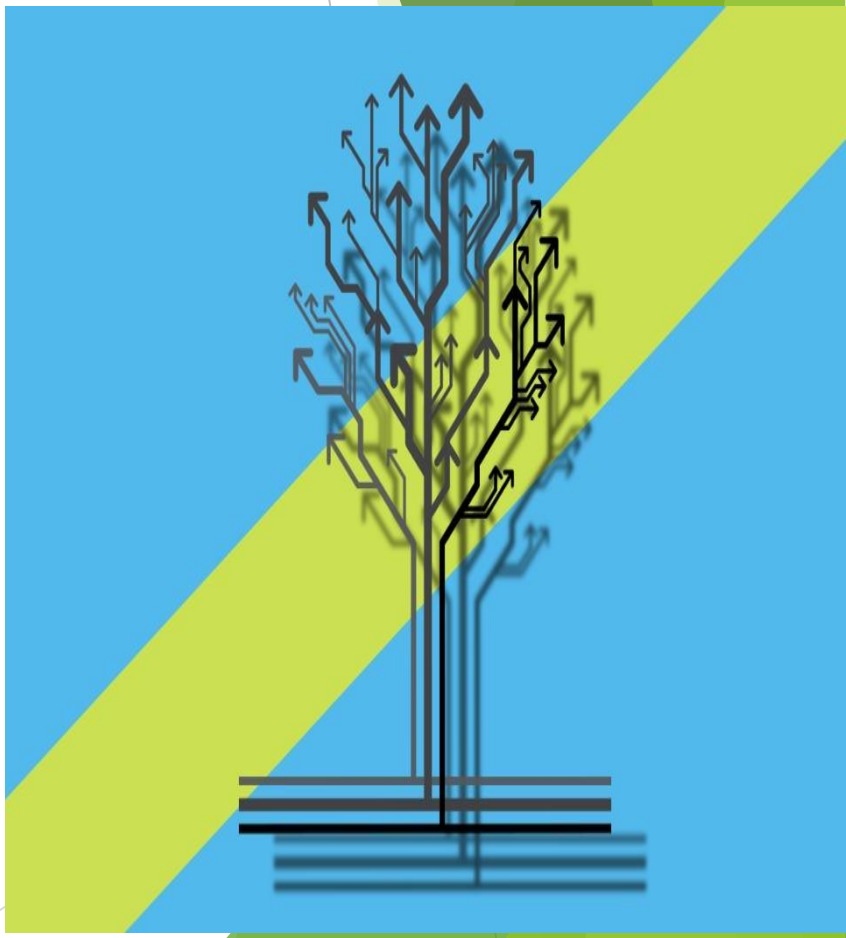
- Karar ağaçlarının belki de en önemli özelliğinden biriside düşük maliyetli olmasıdır.
- Yine karar ağaçları görsel gösteriminden dolay anlaşılır, kolay yorumlanabilir ve veri tabanlarına kolay entegrasyon yapılmaktadır.
- Güvenilirlik bakımından oldukça iyi durumdadır ve bu yüzden yoğun olarak tercih edilmektedir.
- Kullanılan ağaç yapılar görselleştirilebilir.
- Veri hazırlığına çok az ihtiyaç duymaktadır.
- Hem sayısal hem de kategorik veri tipleri ile işlem yapabilmektedir.
- Çok çıktılı problemlere çözüm sunabilmektedir.
- İstatistiksel testler kullanılarak bir modelin doğrulanması mümkündür.

Veri Madenciliğinde Karar Ağaçları

- Karar ağacı karar vericiye birçok avantaj sağlamasına rağmen bir takım **dezavantajları** da bulunmaktadır.
- **Aşırı Uyuma** (Overfitting): Karar ağaçları, eğitim verilerine aşırı uyum sağlayabilir. Özellikle çok derin ve karmaşık ağaçlar, eğitim verilerine tam olarak uymak yerine gürültüyü ve rastgeleliği öğrenebilir, bu da genelleme yeteneğini azaltabilir.
- **Çoklu Sınıflandırıcıları Zor Anlamak**: Karar ağaçları çok karmaşık hale gelebilir, özellikle çok sayıda sınıf veya özellik olduğunda. Bu durumda, ağacın yapısını anlamak ve yorumlamak zor olabilir.
- **Dengesiz Veri Kümesi ile Başa Çıkmak Zor Olabilir**: Karar ağaçları, dengesiz sınıf dağılımlarına sahip veri kümeleriyle başa çıkmakta zorlanabilir. Dengesiz sınıflar, ağacın eğitilmesi ve doğruluğu üzerinde olumsuz bir etkiye sahip olabilir.
- **Duyarlılık**: Küçük veri setleri veya veri setlerindeki küçük değişiklikler, karar ağacının yapısını ve sonuçlarını önemli ölçüde etkileyebilir. Bu da modelin kararlılığını azaltabilir.
- **Eğitim Süresi**: Büyük ve karmaşık veri setlerinde, karar ağaçlarının eğitimi zaman alabilir. Özellikle veri setinde çok sayıda özellik veya örnek varsa, ağacın oluşturulması uzun sürebilir.
- **Anomali ve Gürültülü Verilere Hassasiyet**: Karar ağaçları, anomali veya gürültülü verilere hassastır. Bu tür veriler, ağacın doğruluğunu ve genelleme yeteneğini olumsuz yönde etkileyebilir.

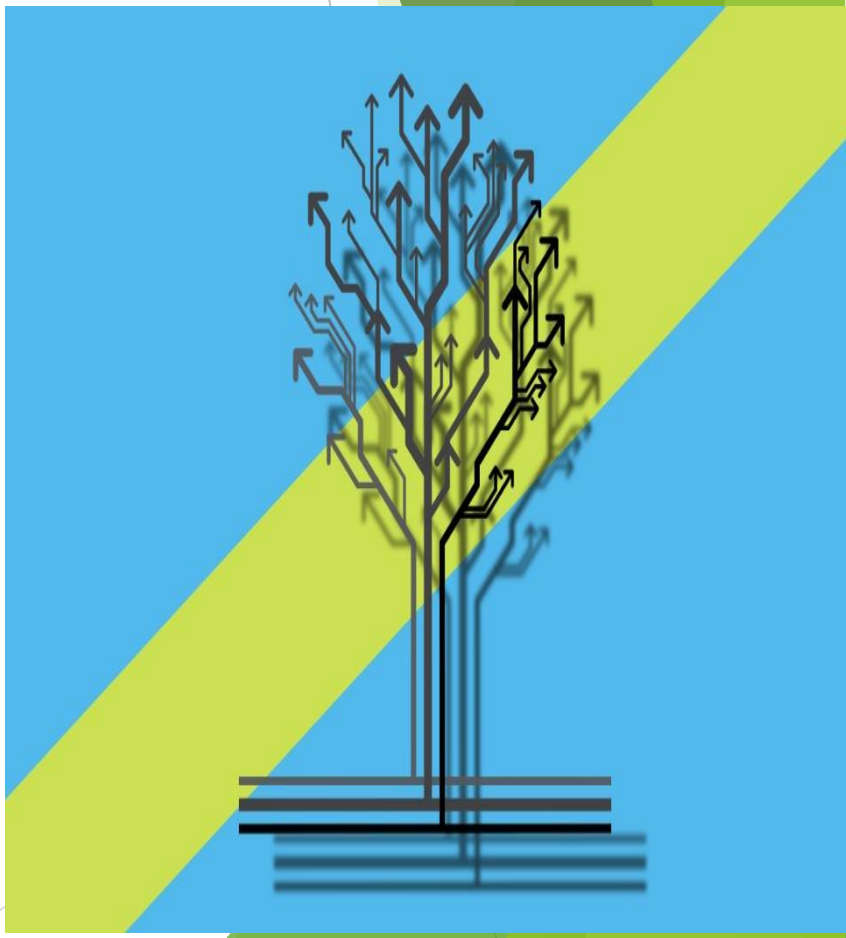
Ağaç Veri Modeli

- Karar ağaçlarında yapı gereği tümevarım söz konusudur. Karar ağaçlarının yapısı düğümler, dallar ve yapraklar olmak üzere üç bölümden oluşmaktadır.
- Düğüm gerçekleştirilecek araştırmayı belirtirken ağacın her bir dalı sınıflama işlemini tanımlamaktadır. Karar ağacı modelinde her bir yaprak dallara dallar ise düğüme bağımlıdır. Karar ağaçlarında işlemler ardışık şekilde gerçekleşmektedir



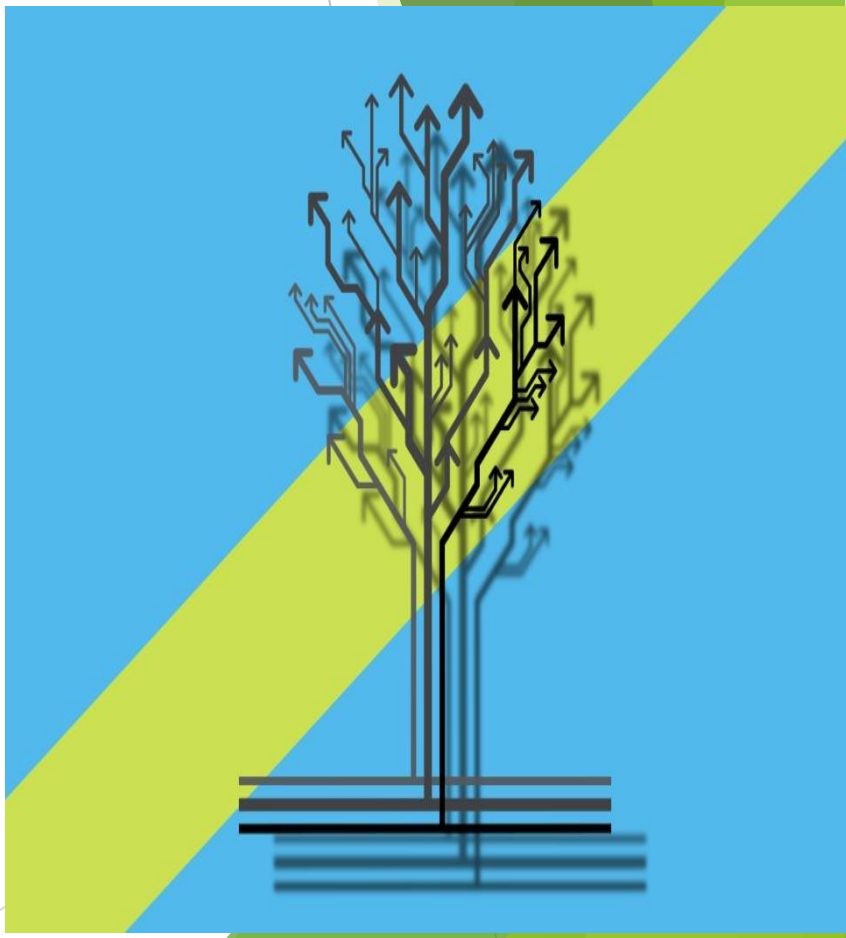
Ağaç Veri Modeli

- Ağaç, bir kök işaretçisi, sonlu sayıda düğümleri ve onları birbirine bağlayan dalları olan bir veri modelidir; aynı aile soyağacında olduğu gibi hiyerarşik bir yapısı vardır ve orada geçen birçok kavram buradaki ağaç veri modelinde de tanımlıdır.



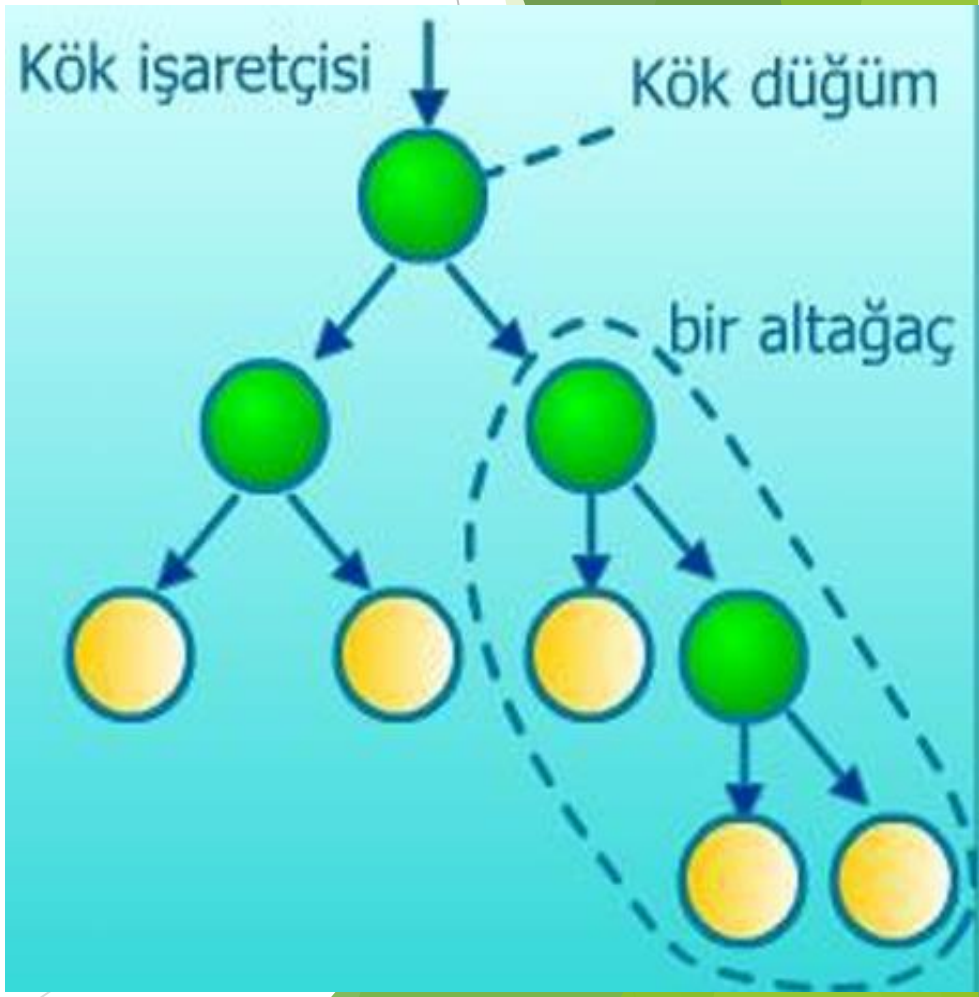
Ağaç Veri Modeli

- Örneğin çocuk, kardeş düğüm, aile, ata gibi birçok kavram ağaç veri modelinde de kullanılır. Genel olarak, veri, ağacın düğümlerinde tutulur; dallarda ise geçiş koşulları vardır denilebilir.
- Her biri değişik bir uygulamaya doğal çözüm olan ikili ağaç, kodlama ağacı, sözlük ağacı, kümeleme ağacı gibi çeşitli ağaç şekilleri vardır; üstelik uygulamaya yönelik özel ağaç şekilleri de çıkarılabilir.



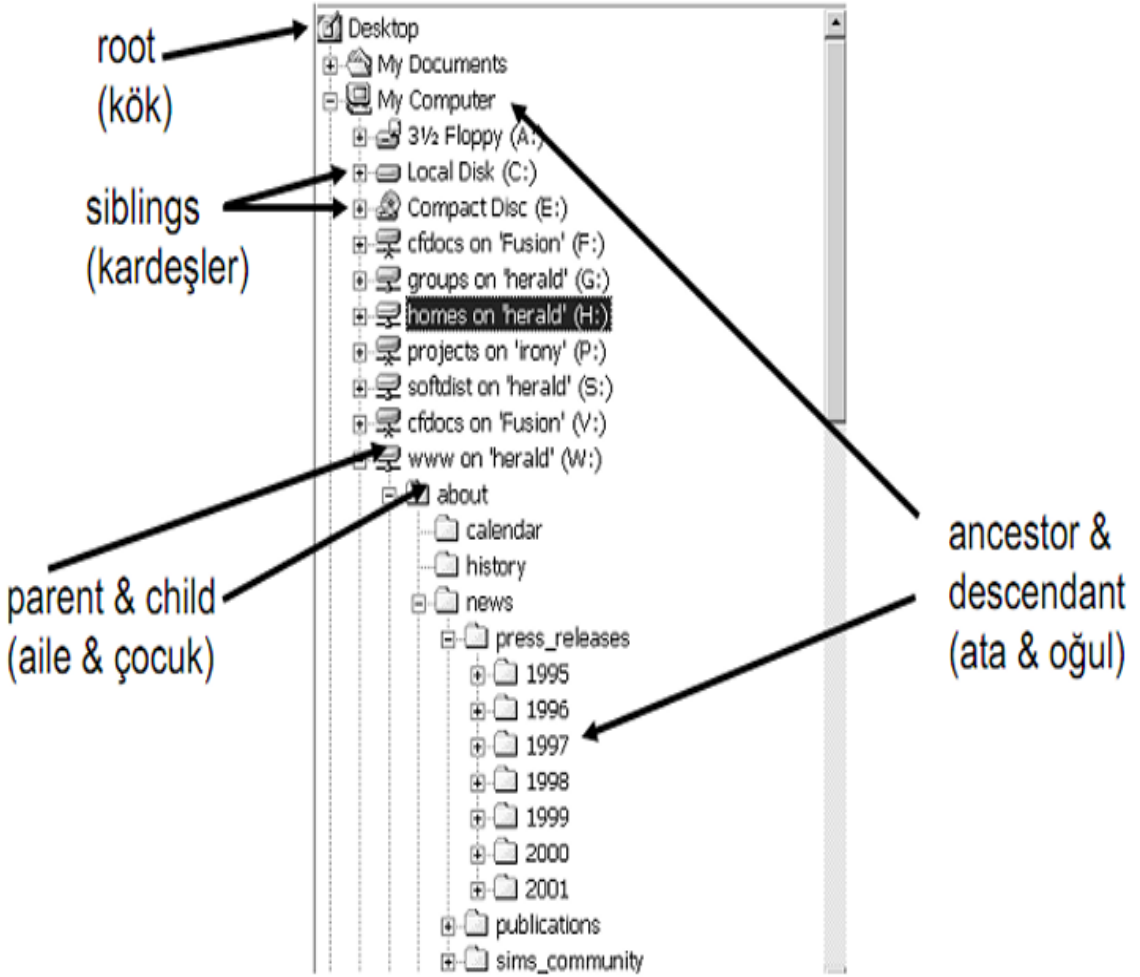
Ağaç Veri Modeli

- Bağlı listeler, yığınlar ve kuyruklar doğrusal (linear) veri yapılarıdır. Ağaçlar ise doğrusal olmayan belirli niteliklere sahip iki boyutlu veri yapılarıdır.
- Ağaçlar hiyerarşik ilişkileri göstermek için kullanılır.
- Her ağaç düğümler(node) ve kenarlardan (edge)oluşur.
- Her bir node (düğüm) bir nesneyi gösterir.
- Her bir kenar (bağlantı) iki node arasındaki ilişkiyi gösterir.
- Arama işlemi bağlı dizilere göre çok hızlı yapılır.

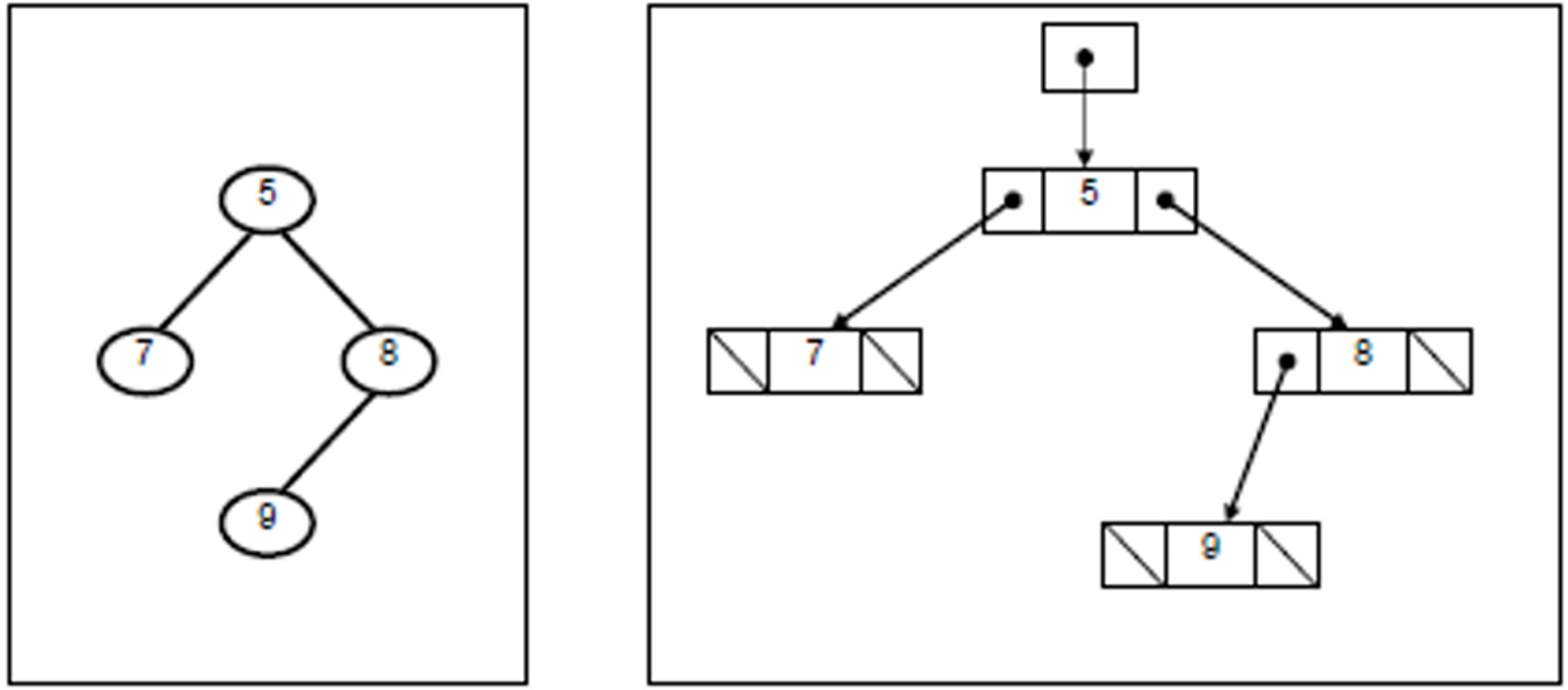


Ağaç Veri Modeli Temel Kavramları

- Ağaçlardaki düğümlerden iki veya daha fazla bağ çıkabilir. İkili ağaçlar (binary trees), düğümlerinde en fazla iki bağ içeren (0, 1 veya 2) ağaçlardır. Ağacın en üstteki düğümüne kök (root) adı verilir.
- Uygulamaları:
  - Organizasyon şeması
  - Dosya sistemleri
  - Programlama ortamları



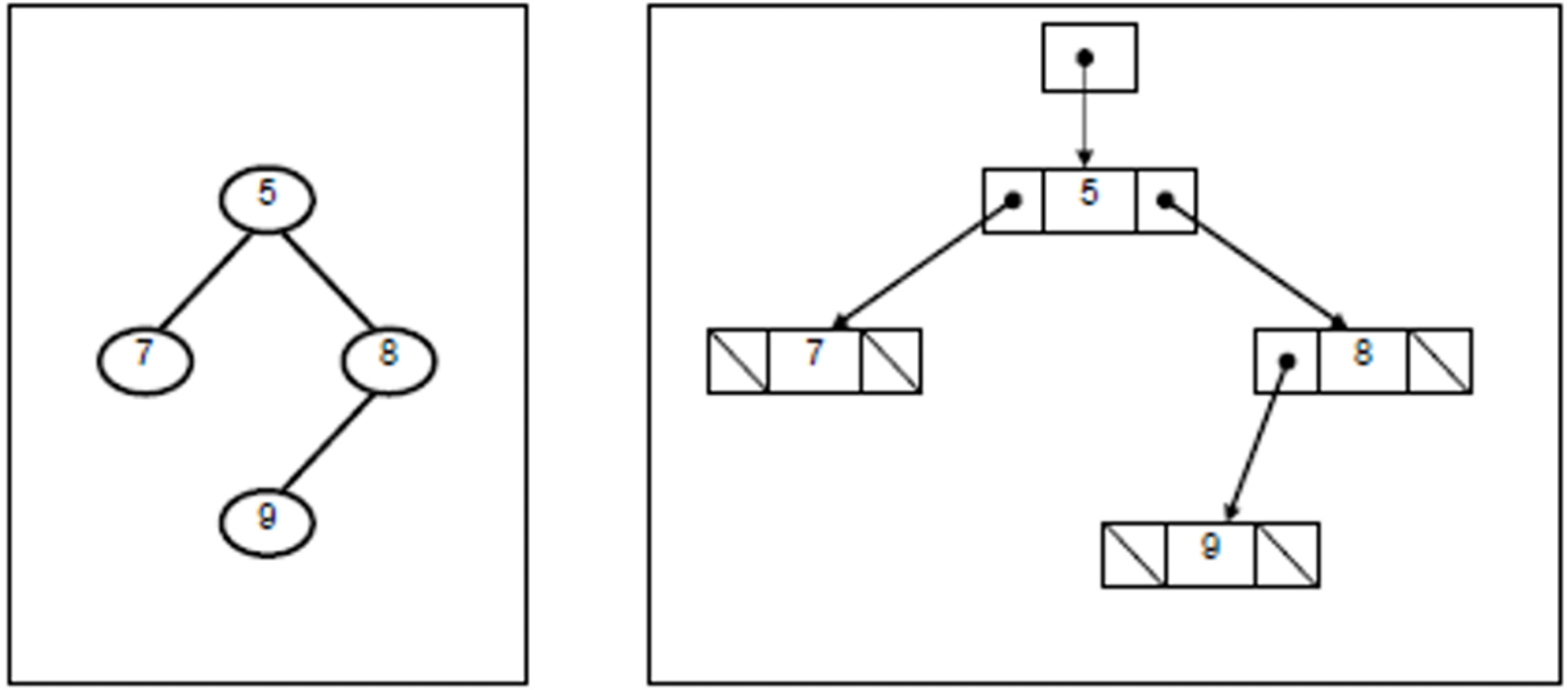
Ağaç Veri Modeli Temel Kavramları



İkili ağacın grafiksel gösterimleri

- Şekilde görülen ağacın düğümlerindeki bilgiler sayılardan oluşmuştur. Her düğümdeki sol ve sağ bağlar yardımı ile diğer düğümlere ulaşılır. Sol ve sağ bağlar boş ("NULL"="/"="\") da olabilir.

Ağaç Veri Modeli Temel Kavramları

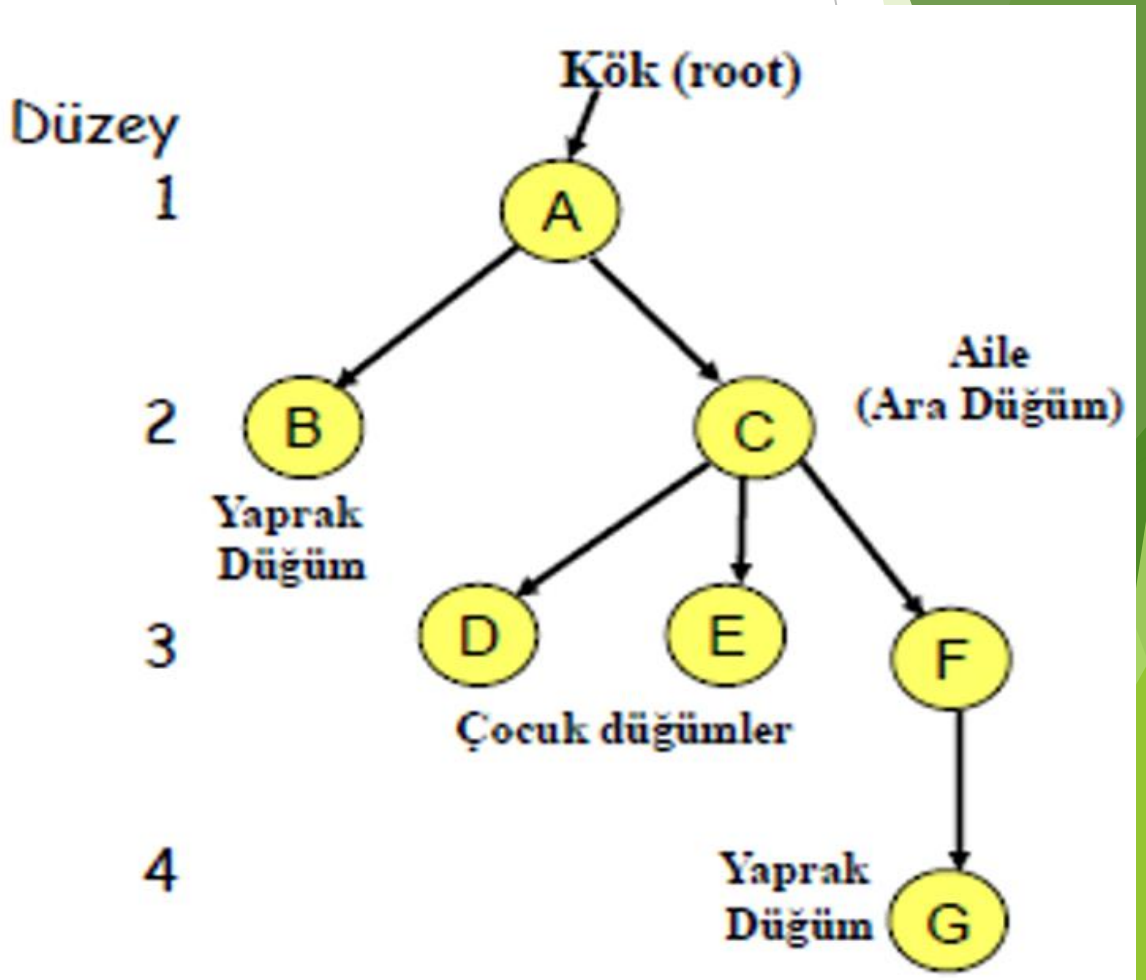


İkili ağacın grafiksel gösterimleri

- Düğüm yapıları değişik türlerde bilgiler içeren veya birden fazla bilgi içeren ağaçlar da olabilir. Doğadaki ağaçlar köklerinden gelişip göğe doğru yükselirken veri yapılarındaki ağaçlar kökü yukarıda yaprakları aşağıda olacak şekilde çizilirler.

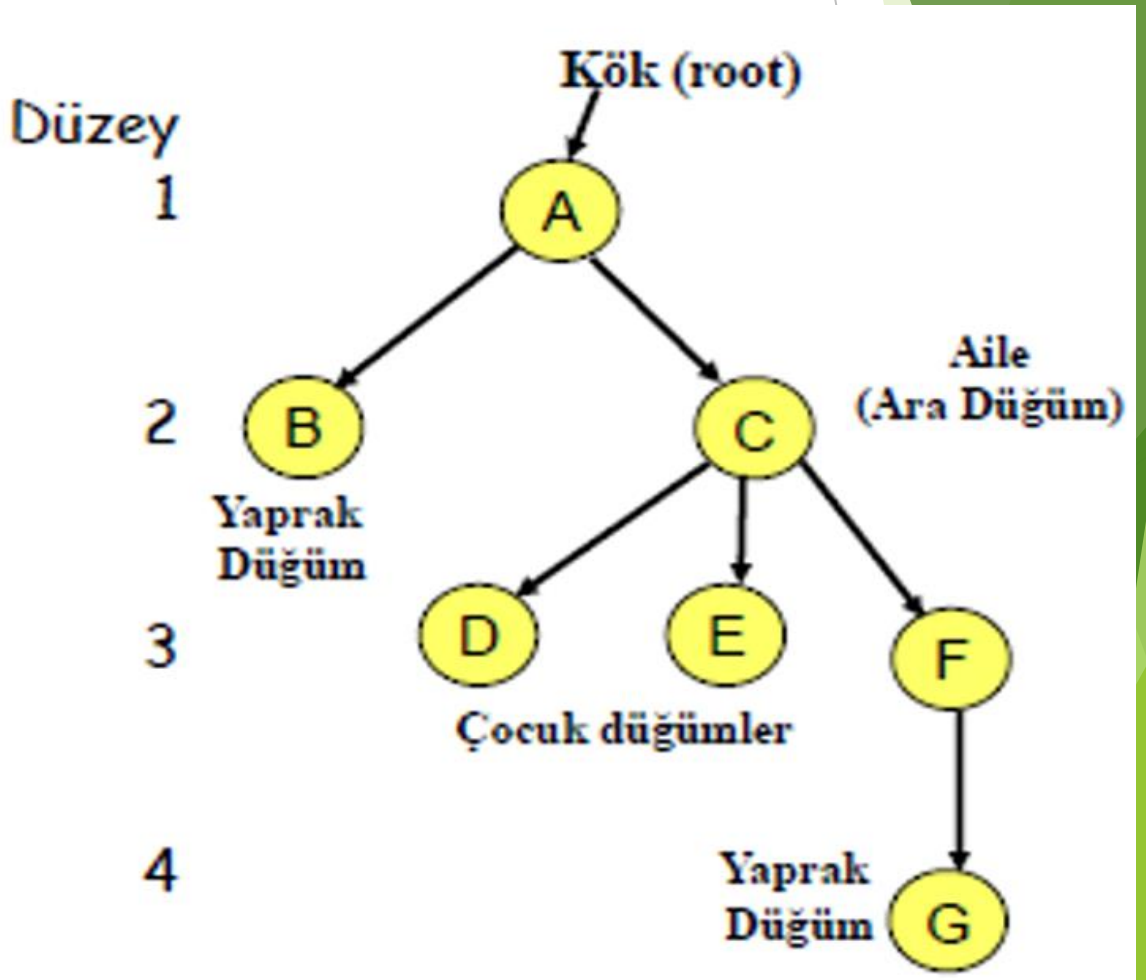
Ağaç Veri Modeli Temel Kavramları

- Şekildeki ağaç, **A düğümü kök olmak üzere 7** düğümden oluşmaktadır. Sol alt ağaç B kökü ile başlamakta ve sağ alt ağaç da C kökü ile başlamaktadır. A'dan solda B'ye giden ve sağda C'ye giden **iki dal** (branch) çıkmaktadır.



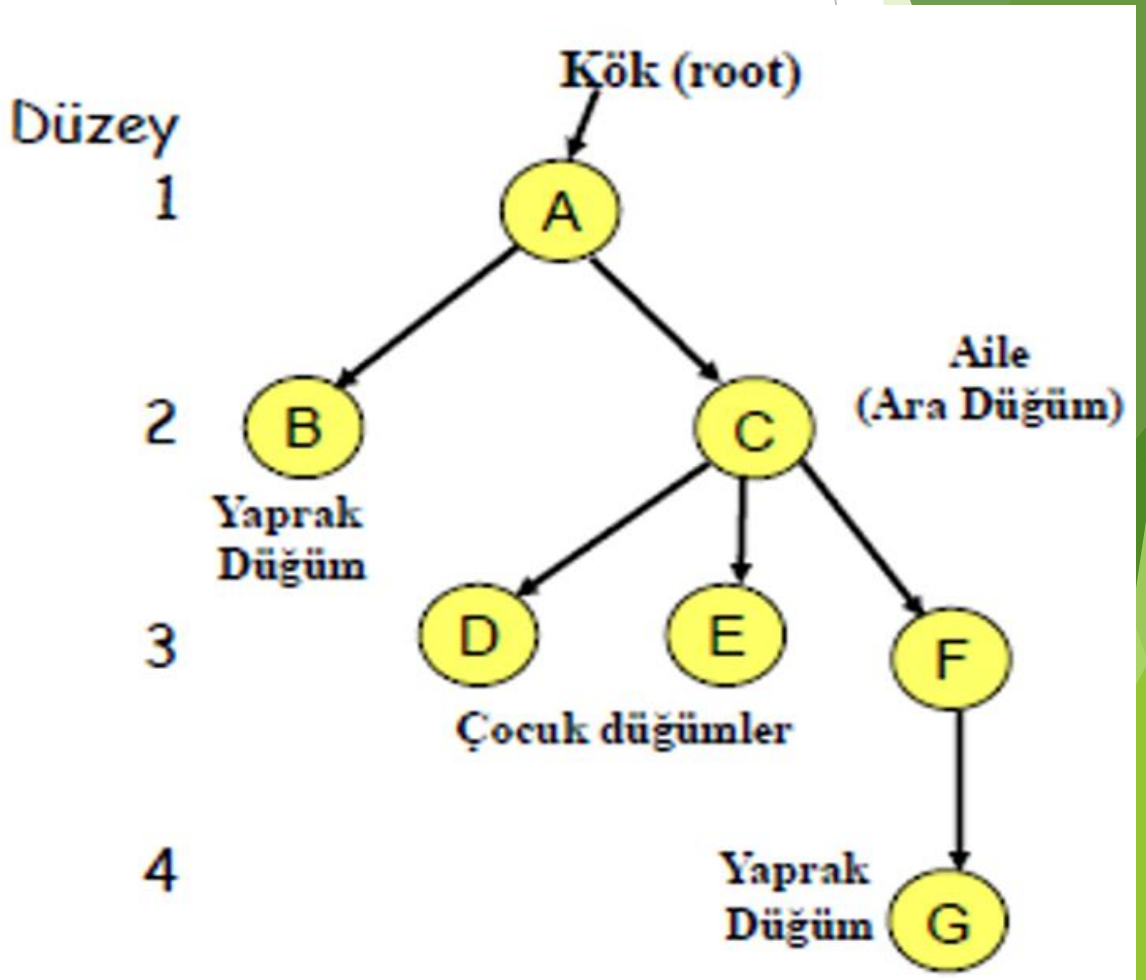
Ağaç Veri Modeli Temel Kavramları

- Düğüm (Node)
  - Ağacın her bir elemanına düğüm adı verilir.
- Kök Düğüm (Root)
  - Ağacın başlangıç düğümüdür
- Çocuk (Child)
  - Bir düğüme doğrudan bağlı olan düğümlere o çocukları denilir.
- Kardeş Düğüm (Sibling)
  - Aynı düğüme bağlı düğümlere kardeş düğüm veya kısaca kardeş denir.



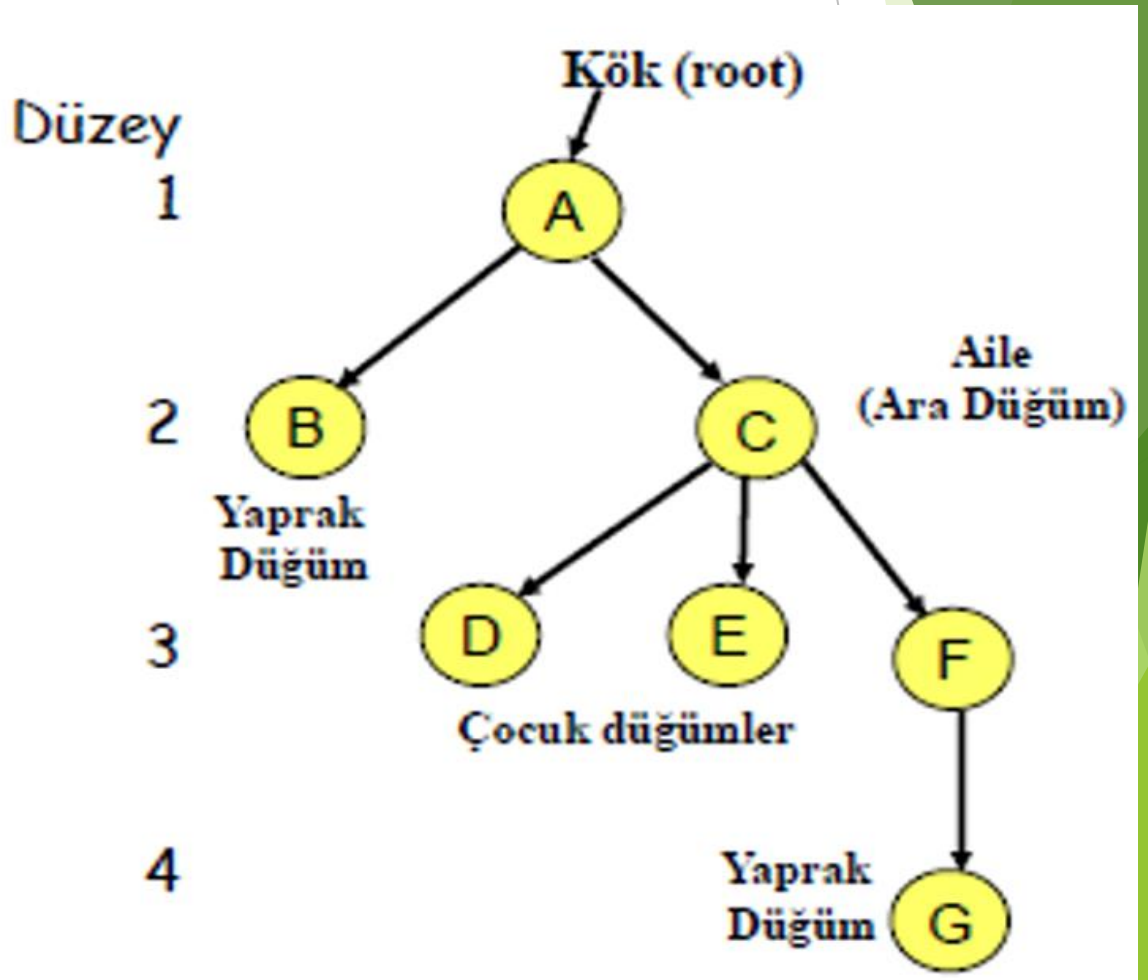
Ağaç Veri Modeli Temel Kavramları

- Aile(Parent)
  - Düğümlerin doğrudan bağlı oldukları düğüm aile olarak adlandırılır; diğer bir deyişle aile, kardeşlerin bağlı olduğu düğümdür.
- Ata (Ancestor) ve Torun (Dedscendant)
  - Aile düğümünün daha üstünde kalan düğümlere ata denilir; torun, bir düğümün çocuğuna bağlı olan düğümlere denir.



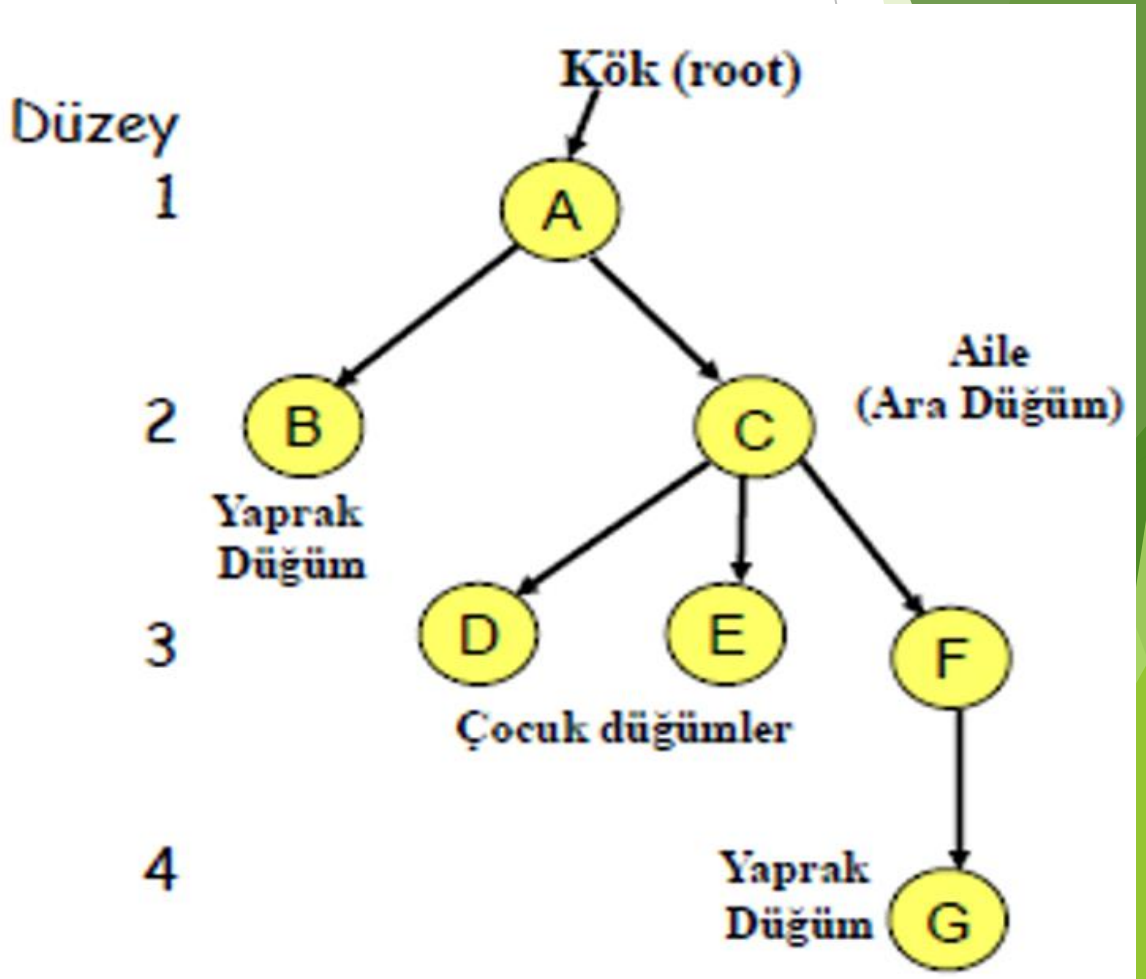
Ağaç Veri Modeli Temel Kavramları

- Derece(Degree)
  - Bir düğümden alt hiyerarşiye yapılan bağlantıların sayısıdır; yani çocuk veya alt ağaç sayısıdır.
- Düzey(Level) ve Derinlik(Depth)
  - Düzey, iki düğüm arasındaki yolun üzerinde bulunan düğümlerin sayısıdır. Kök düğümün düzeyi 1, doğrudan köke bağlı düğümlerin düzeyi 2'dir. Bir düğümün köke olan uzaklığı ise derinliktir. Kök düğümün derinliği 1'dir.
- Yaprak(Leaf)
  - Ağacın en altında bulunan ve çocukları olmayan düğümlerdir.

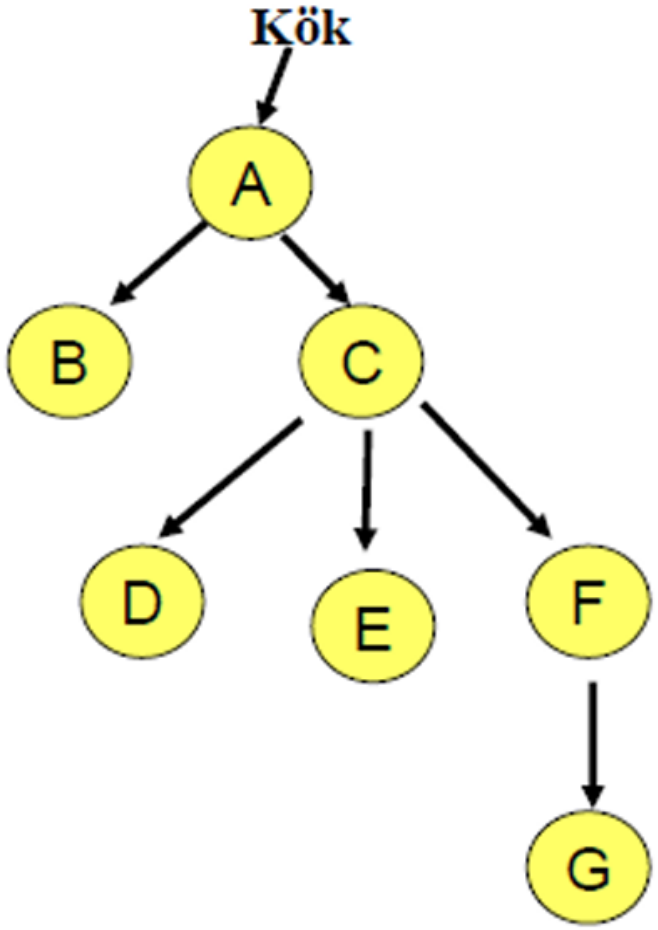


Ağaç Veri Modeli Temel Kavramları

- Yükseklik(Height)
  - Bir düğümün kendi silsilesinden en uzak yaprak düğüme olan uzaklığıdır.
- Yol(Path)
  - Bir düğümün aşağıya doğru (çocukları üzerinden) bir başka düğüme gidebilmek için üzerinden geçilmesi gereken düğümlerin listesidir.

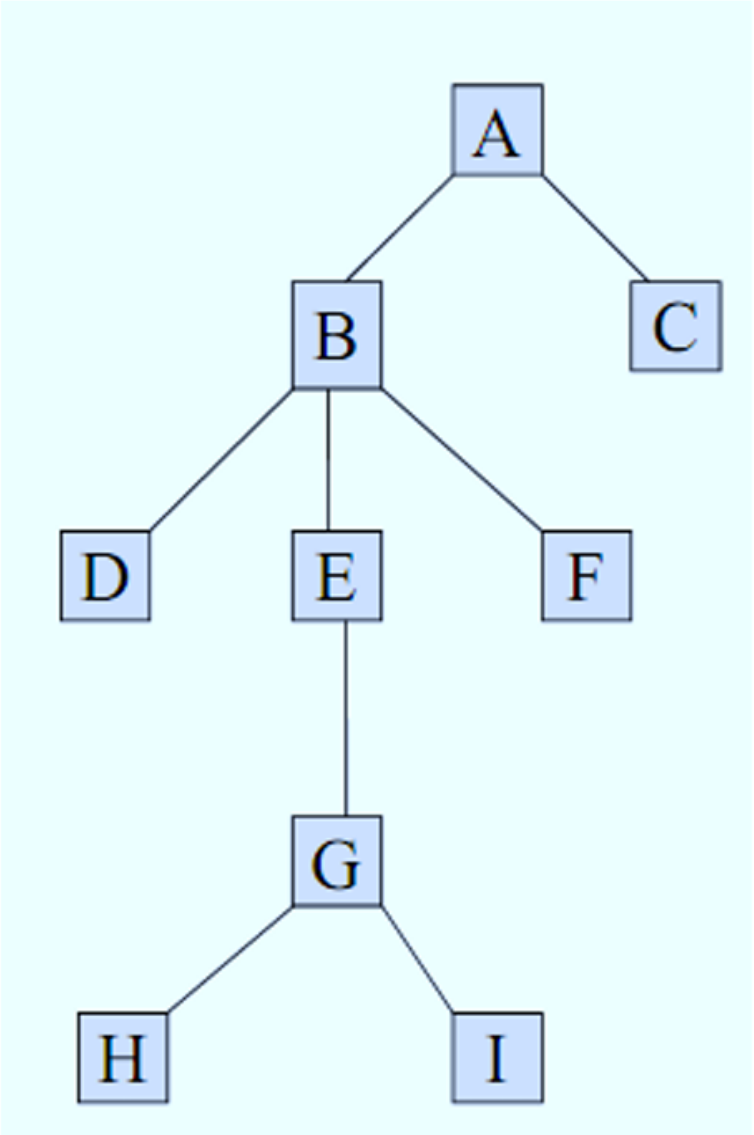


Ağaç Veri Modeli Temel Kavramları



Tanım	Kök	B	D
Çocuk	2	0	0
Kardeş	1	2	3
Düzey	1	2	3
Aile	yok	kök	C
Ata	yok	yok	kök
Yol	A	A,B	A,C,D
Derinlik	1	2	3
Yükseklik	3	2	1

Ağaç Veri Modeli Temel Kavramları



Tanım	Değer
Düğüm Sayısı	9
Yükseklik	4
Kök Düğüm	A
Yapraklar	C,D,F,H,I
Düzey Sayısı	5
H'nin Ataları	E,B,A
B'nin Torunları	G,H,I
E'nin Kardeşleri	D,F

Ağaç İşlemleri

- **Ağaç Oluşturma:**
  - Düğüm (Node) sınıfı tanımlama: Ağaç yapısındaki her düğümü temsil etmek için bir sınıf tanımlanabilir. Bu sınıf genellikle düğümün değerini ve çocuk düğümleri listesini içerir.
  - Ağacı dict (sözlük) olarak oluşturma: Ağacın kök düğümünü temsil eden bir sözlük oluşturulabilir. Bu sözlük, her bir düğümün anahtar olarak ve alt düğümleri değer olarak içerebilir.
- **Gezinme (Traversal):**
  - Öncelik sırası (pre-order), İkilik sırası (in-order), veya Sonrası sırası (post-order) gibi farklı gezinme yöntemlerini uygulama: Bu yöntemler ağacı belirli bir sıra ile gezerek düğümleri ziyaret etmeyi sağlar.
  - Derinlik-öncelikli arama (DFS) veya Genişlik-öncelikli arama (BFS) gibi farklı arama yöntemlerini uygulama: Bu yöntemler, ağaç yapısını belirli bir düğümü bulmak veya gezinmek için kullanılır.

Ağaç İşlemleri

- **Düğüm Ekleme ve Silme:**
  - Yeni bir düğüm eklemek: Ağaç yapısına yeni bir düğüm eklemek için uygun bir ekleme algoritması kullanılabilir. Bu, belirli bir konumda veya belirli bir düğümün altında yeni bir düğüm eklemeyi içerebilir.
  - Bir düğümü silmek: Belirli bir düğümü ağaç yapısından silmek için uygun bir silme algoritması kullanılabilir. Bu, düğümün kendisi ve ona bağlı olan alt ağacı kaldırmayı içerebilir.
- **Ağacı Dolaşma (Tree Traversal):**
  - Öncelikle (pre-order), Orta (in-order), Sonra (post-order) gibi farklı dolaşma yöntemlerini uygulama: Bu yöntemler, ağacın düğümlerini belirli bir sırayla ziyaret etmek için kullanılır. Örneğin, in-order dolaşma, ağacın sıralı bir şekilde gezilmesini sağlar.
- **Ağacı Görselleştirme:**
  - Görselleştirme kütüphanelerini kullanarak ağacı görselleştirme: Matplotlib, Graphviz veya NetworkX gibi kütüphaneler, ağaç yapısını görselleştirmek için kullanılabilir. Bu, ağacın yapısını ve içeriğini daha iyi anlamak için faydalı olabilir.